

Revision	File Name	Changelist	Date	Description
----------	-----------	------------	------	-------------

6218	//depot/m...	6218	2000/01/27	updated to use globally monotonically increasing sequence numbers
------	--------------	------	------------	---

6181	//depot/m...	6181	2000/01/27	code to control host and device sequence numbers.
------	--------------	------	------------	---

```

#include "storage.h"
#include "CalSequence.h"
#include "caldb.h"

5  SequenceNumbers g_minSequenceNumbers;

class iCalBookmarkFile: public PersistentStore
{
public:
10     iCalBookmarkFile()
        : PersistentStore("Calendar Sync Bookmark", &g_minSequenceNumbers, sizeof(SequenceNumbers))
        {
            load_data();
        }

        virtual bool backup_restore_completed()
        {
            load_data();
            return true;
        }
20     };

    iCalBookmarkFile iCalBookmark;

void
FindLastDeviceSequence()
{
    // find the largest device sequence number
    unsigned int nMax = 0;

    DbRecordHandle dbHandle;

35     if ((dbHandle = Calendar::calendarDatabase.get_first_record()) != DB_NULL_HANDLE)
    {
        do
        {
            Calendar::CalendarRecord cr(dbHandle);
            unsigned int nTest = cr.GetDeviceSequenceID();
            if (nTest > nMax)
                nMax = nTest;
        } while ((dbHandle = Calendar::calendarDatabase.get_next_record(dbHandle)) != DB_NULL_HANDLE);
40     }

```

```

45         if (!nMax)
            nMax++;

        if (nMax > g_minSequenceNumbers.lastDeviceSequence)
        {
            g_minSequenceNumbers.lastDeviceSequence = nMax;
            iCalBookmark.save_data();
        }
    }

55     unsigned int
    GetNextDeviceSequence()
    {
        // Disaster recovery. If we can't find a last sequence number, go through the DB and guess.
        if (!g_minSequenceNumbers.lastDeviceSequence)
            FindLastDeviceSequence();

        unsigned int nNext = ++g_minSequenceNumbers.lastDeviceSequence;

        iCalBookmark.save_data();
        return nNext;
    }

65     }

    unsigned int
    GetMinHostSequence()
    {
        return (g_minSequenceNumbers.minHostSequence);
    }

70     }

    void
    SetLastHostSequence(unsigned int seq)
    {
        if (seq > g_minSequenceNumbers.lastHostSequence)
        {
            g_minSequenceNumbers.lastHostSequence = seq;
            iCalBookmark.save_data();
        }
    }

80     }

    void
    SetLastDeviceSequence(unsigned int seq)
    {

```